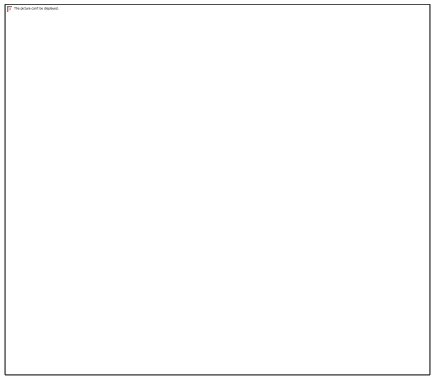

ASHESI UNIVERSITY COLLEGE

MHEALTH AND MOBILE SECURITY

Applied Project

B.Sc. Computer Science

Nathan Yaw Oppong Donkor

2017

ASHESI UNIVERSITY COLLEGE

MHealth And Mobile Security

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi University College in partial fulfilment of the requirements for the award of a Bachelor of Science degree in Computer Science

Nathan Yaw Oppong Donkor

April 2017

DECLARATION

I hereby declare that this Applied Project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

…….

Candidate's Name:

…….

Date:

…….

I hereby declare that preparation and presentation of this Applied Project were supervised in accordance with the guidelines on supervision of Applied Projects laid down by Ashesi University College.

Supervisor's Signature:

……..

Supervisor's Name:

……..

Date: ……..

Acknowledgement

I wish to express my utmost gratitude to the Almighty God for his blessing, direction and insight that has guided me through this project successfully.

To the one who always believed in me despite everything and continued to encourage me to work hard - my mother, I express my grotesque love and gratitude. Thank you for the support.

My appreciation also extends to my supervisor, Mr. Kwadwo Gyamfi Osafo-Maafo for his time, enlightenment, guidance and significant contribution to the work done on this project.

To Duki, Nakoh, Roslynne, Pinamang, Anna, and Melissa, I express a great amount of gratitude for the assistance and emotional support you offered to me in the closing stages of my work.

The work done on this project could not have possible without the help of the first developers on the projects, and those who gave me insight on how to go about this project. The success of this project is all credited to you.

To all who contributed to the success of my work, I express my love and gratitude to you and pray God blesses you. This project would not be successful without your support and encouragement.

Abstract

This Applied Project reviews, designs and implements security for the mobile application used by the MHealth project. The result of improving security brings the project in line with Ghana Ministry of Health's Legal and Policy Framework for Health Information and Health Data Reporting. Information security is one of the prime concerns for healthcare nowadays, because even though electronic record keeping enhances efficiency, it also poses a vulnerability of access.

Security is improved in MHealth by encrypting the database mainly and implementing various user authentication techniques. This will enable the application and the project protect their data and prevent unauthorized access by attackers.

Given that Ghana is now moving into a phase of electronic record keeping, this contribution will be a necessary start for the implementation of an electronic system.

Table of Contents

List of Abbreviations

CHPS	Community-Based Health Planning and Services
https	Hyper Text Transfer Protocol Secure
OPD	Out Patient Department
CHO	Community Health Organization
SSH	Secure Shell
STTP	Secure Token Transfer Protocol

Chapter 1: Introduction

1. Background Context of the Topic

This project builds on an already existing product by reviewing and implementing mobile security within the MHealth mobile application. The MHealth project is currently used by nurses in the field, who use mobile phones or tablets to improve the efficiency level of their work. Initially, health records for this project were recorded in hard copy, and this made record-keeping hectic. The introduction of the project's mobile application (named '*Yaresa'*) aided in reducing the hectic nature of record-keeping. Despite aiding in the efficiency of the project, it is important to review and increase the security of the mobile application and data located on the smart phones used by the nurses. Considering this, this applied project is going to tackle the security of this mobile application by achieving some objectives.

1.2 MHealth

MHealth is a project which was started in Ashesi, and involves nurses who offer health services in rural areas. Their primary responsibility is to provide health care by recording the credentials of patients in rural areas, taking the information obtained to doctors in hospitals so that these doctors review the information and then diagnose and prescribe medication for these patients. The process of record-keeping was however hectic because of the method by which nurses spent a lot of time preparing reports.

There were two steps for preparing reports. First, tally books were kept and recorded by hand. Secondly, a paper report had to then be hand–delivered a distance away from where some of the nurses worked. Instead of manual tallying and reporting, data is collected with mobile tablets which have the application installed on them. The MHealth application also performs various tallies and provides data for reporting purposes Their current work is to bring statistics to the district office where they track and know what is going on in the field. Nurses

can either use phones to help clients fill out their forms, or use phones to send data over the Internet to a server for reports.

1.3 Mobile Security

The scope of this project involves protecting both health data and applications stored on and transmitted from smartphones, tablets, laptops and other mobile devices. Mobile security has a wide spectrum that encapsulates a lot of fields, ranging from protecting physical mobile devices from malware threats to securing mobile devices and their data in the case of theft, unauthorized access or accidental loss of the mobile device (Webopedia Definition, n.d.). Looking at this, there are various ways of achieving mobile security. Some of these include:

1. Access Control
2. Database Security
3. Physical Device
4. Network Security

1.4 MHealth And Mobile Security

When we talk about health, we do not necessarily talk about just getting treated with medication. Healthcare also involves the privacy of data that is stored for record-keeping, analysis, research and the safety of health data from being tampered with. There have been cases of health security in recent years. In 2016, Russian hackers leaked information of some Rio Olympic Athletes, these included Simon Biles, Venus Williams and Serena Williams. This leak showed these athletes had some illnesses they found embarrassing and thus had not told the public yet. They found this very embarrassing and they had to come out to explain they were

under medication (Rumsby,2016). This illustrates how confidential health information is, and thus shows how critical it is to protect such information.

This poses a significant problem because there has been no approach to the security concern of MHealth yet, hence “Yaresa” which is the name of the mobile application. In light of this, an approach skewed toward security must be implemented to prevent any unfortunate occurrences concerning the project. Unfortunate in the sense that, people can lose valuable information which can hurt some individuals who have trusted the credibility and integrity of the project. When the security requirements of the MHealth application are met, the credibility of the program will be increased, and this will attract more clients. This will also contribute to the field of Mobile Health Security. Aside contribution, we will also enhance security for the MHealth program.

According to the Ministry of Health, there are certain rules and regulations that have been put in place to safeguard against the unfortunate case of data tampering, data loss or data theft or corruption. These have been documented in a document titled ‘Legal and Policy Framework for Health Information and Health Data Reporting’. This document outlines liabilities and sanctions as part of the medical records policy on protecting health information. An excerpt of this document which details the ministry’s policies on the exchange of health information is listed in the appendix. *(Appendix 1)*

1.5 Status of M-Health Application

The MHealth for CHPS project developed a mobile application for health, which is now being used by number of CHPS zones. The health officers use this application to track OPD cases, immunization and family planning. The application allows health officers to plan and send reports to the management team at the district level. The MHealth administration have developed a very good relationship with the health services of two districts, and the community

health officers (CHOs) in the district that we work in. We have thus established a channel for students to create mobile applications and other tools to be used by community health officers.

These are the main functions of the MHealth project that have been addressed since is development.

Reporting and Data Management: The health officers who use the application confirm that the amount of time and effort they devote to compiling the monthly reports has been cut significantly. The application generates summary reports on OPD case attendance, immunization and family planning which health officers use to complete their report. Without the application, they must compile these reports manually. This is a waste of time because the time spent on compiling reports manually can be spent on attending to other health issues. Sometimes when community members misplace their OPD cards, the health officers use the application to retrieve their OPD files, and verify insurance information. We have developed the application to a point where health officers can submit information electronically to the district.

Planning: Using this application, health officers operating in a CHPS zone can view how many children need to be vaccinated each week or month to estimate the volume of vaccination to prepare. They can also compile a list of children scheduled for vaccination to remind their parents. One challenge in the health service is that NHIS members don't renew their subscription on time. The application thus prompts health officers with the list of people whose insurance will expire within 3 months. They can use this information to prompt the clients to renew their insurance during home visits, outreach and other opportunities.

Other Uses: Health officers have found creative ways of using the tablet PCs that they have been provided with. They take pictures of activities they have done, to report and share. We have seen one health officer who took pictures of a skin disease to report to his management team through the application.

The initial module of the application which was tested with few CHPS zones is the knowledge module. This module allows health officers to access training videos using peer-to-peer networks, hence avoiding the need for Internet. The health officers who take up community health officer posts are young nurses with less than 3-year experience. Their continuous professional development can be supported by providing them videos with new knowledge. We tested the concept by preparing a set of training videos on Mental Health Law with the help of the Student Drama Club at Ashesi and distributing it to health officers using the tablet. During interviews carried out later, the health officers recalled information from the video on how to engage metal health patients. The feedback on the method was that they can take advantage of such resource. They also suggested that they can use the videos to educate the community members.

1.6 Current and Further Work

The mobile application has been continuously developed in the following ways, using constructive and guided feedback collected from health officers who use the mobile application.

1. The interface has been improved to make the application a useful tool for health officers not only in reporting to their management team but also in aiding them to plan their daily work, as this makes them more efficient.
2. A module is in the pipeline that is going to aid compile NHIS insurance claims using the application. This process of preparing NHIS claims is a hurdle that takes a lot of the health officers time.
3. Apart from the significant work done by some of the Ashesi faculty members, two final-year students at Ashesi have developed additional modules for data management and professional training as part of their final year project. In addition to that, a group of third year students at Ashesi developed a module as part of their class project,

that enables district management teams assign tasks and targets to health officers posted in CHPS zones throughout the districts.

4. With the district health services, we will like to develop training multimedia materials around specific topics identified by health officers and their management team. One topic suggested by the management team of West Gonja is how to handle adverse effects of vaccination. We also want to train health officers on how to use data for their work, because we observed that the health officers collect a lot of data but do not use it for their work. With a bit of training and the right tools, each health officer can be much more effective.

At the end of this chapter, there are a few things that stand out as a contribution to the work that is going to be accomplished by the end of the project**.** To start with, we noticed that no one has approached this project from the security point of view. The things that need security include the application data and the database that has all the tables that will aid in storing all necessary records that will continue to contribute as effective record keeping for the project's mobile application.

In summary, the security of the ***"Yaresa"*** application is paramount to the thriving of the MHealth project and will contribute to the health sector in various ways. MHealth is a project that started in Ashesi and with the help of some students from the school, some people developed an application to aid record keeping and efficiency for the project. In relation to the project, mobile security is a field of the mobile application that has not been considered. Mobile security has a wide spectrum, but in this project, attention will be given to just a few areas of mobile security. Some of these fields include database encryption, access control and password authentication. The Ministry of Health has also developed paradigms to guide any entity that wants to take part in electronic record keeping. This will help safeguard the medical records of individuals and contribute to the integrity and protection of data. Because of this, there is realization on how mobile security is fundamental to the preservation and authenticity of the

"***Yaresa***" mobile application. All data that interacts with this mobile application must be secure for the application to serve the needs of its patients and secure their data.

Chapter 2: Related Work

2.1 Literature Review

In this chapter, there will be discussions about various literature and how they have influence the approach used for this project. The various literature that influenced the approach include mobile security, encryption, two factor authentication and security measures for mobile applications.

Sadeghi, in his article "Mobile security and privacy: The quest for the mighty access control" talks about how smartphones are changing the lives of people and the society we live in (Sadeghi, 2013). Mobile applications are utilized in every aspect of our lives to extend the efficiency of everything. This is seen in our everyday lives with critical services such as online banking, health records, enterprise applications and social networking. With this arising interest and introduction of mobile applications into human lives, people want to find vulnerabilities to these things. As a result, there is a need to protect these applications that affect the lives of majority of the populace (Sadeghi, 2013).

There is general concern to secure all entities that communicate to facilitate the existence of these mobile applications, from the hardware to the operating system to the software. There is a need to provide access tokens which aid prevent unauthorized access to various mobile applications. The journal article elaborates on the need for security in mobile applications since they seem to be the main point of vulnerability. The article talks about various threats to mobile applications and how their data can be compromised. It then moves forward to discuss the Android-based applications because of their open-source and popular nature. After investing a significant amount of literature on Android security, trends were observed and measures were proposed to addressed security concerns like access control to the user's private data. It also addresses the fact that security might be user-centered, and provides solutions to various security threats that may compromise data. Lastly, the article talks about various entry points of breaches for the Android operating systems (Sadeghi, 2013).

In "Privacy and data protection in smartphone messengers" (Rottermanner et al.,2015), the security of mobile messengers is discussed as a source of privacy breach, leading to data theft which results in a whole level of social engineering to gain information on individuals. It also discusses privacy concepts on the transmission level that concerns the support for encrypted communication, and discusses permissions granted to mobile applications to use various components and resources on one's mobile phone. These are all explored in the article to serve as a point of reference in discussing and tackling various security points, ranging from hardware to software to the network and finally the human. The article also talks about device theft, which poses as a threat to user privacy. As a result, if a device is stolen, the user may have their data compromised, tampered with or totally lost. Hence, we must prevent against the access of the data when a device is stolen. (Rottermanner et al.,2015)

Another highlight from research was new methods by which some mobile applications are preventing data theft and integrity. One new thing that Whatsapp is doing is end-to-end

encryption. This is where the data is encrypted before it is sent, then eventually decrypted when it gets to the recipient. This is also being implemented by Signal, an instant messaging application. (Muellar et al.,2014)

Applicure.com also suggests some measures that will aid in mobile security. These include separating the database and web servers, encryption of stored files, encryption of backups, keeping patches current, avoiding the use of third party applications, avoiding the use of shared servers and enabling security controls. (Applicure.com, 2016)

Evaluating this insight provided the necessary enlightenment needed to proceed to the next stage of the project. This section tackled several significant highlights that must be addressed in this project. Basically, the most important aspect of mobile applications is their data. This brings us to information security, which involves protecting data at all costs. Using this insight, the requirements or expectations of this project will be detailed in relation to mobile security, in the next chapter.

2.2 Plan for Requirements

In this section, a plan will be devised to develop the functional and non-functional requirements of this project. Hence, the following bullet points will explain each plan for gaining the requirements.

1. Review of existing Documentation - This will delve in to already addressed methods by previous people who have worked on the project in previous years. This will facilitate the reconnaissance of the project to know what work/ contribution already exists in relation to mobile database security.

2. Review of Similar Projects - This will aid in reviewing various methods that have been implemented by others relating to Health and Mobile database security. This will allow me to gain inspiration on various methods to secure databases. This will also allow me to suggest various methods to my client to ensure I meet their needs.

3. Preparation of questionnaire - After this, a questionnaire will be drafted for clients, to receive feedback for user requirements.

4. Submission of questionnaire to clients.

5. Tightening of questionnaire.

6. Gathering requirements from final questionnaire.

Chapter 3: Requirements

3.1. Procedure for Gaining Requirements for the Project

Dealing with the Health project, it was quite difficult to use the traditional approach to gain the requirements of the project. This was because of the aspect of the MHealth project we were dealing with. Traditional approach here refers to issuing questionnaires to the public. This would not have worked since mobile security is a specialty that needs expert opinion or at least someone who is involved in the development of the mobile application.

Since the issue being dealt with is security, the users were limited to the people who built this application and appreciate mobile security. This included a few faculty in Ashesi, who suggested ideas on what to do about the security of the application. Because of this, short interviews were conducted to gain the necessary information on the subject matter. The questions asked during the interviews are listed in the appendix. *(Appendix 2)*

Also, with the help of blogposts, course material for mobile security and MIT

Opencourseware, I could gain some expectations of mobile security. After reviewing the feedback obtained from the interviews and research, some the functional requirements included:

1. Encryption of stored files
2. Encryption of backup
3. Keeping patches current
4. Avoiding the use of third party applications
5. Avoiding the use of a shared server
6. Enabling security controls
7. Encryption of transmission routes
8. Enhancing user authentication
9. Implementing two factor authentications of the user

Also, if any of these requirements should be met, there will be non-functional requirements that will have to be met. These include:

1. Data Integrity
2. Security
3. Performance (Response time)
4. Stability
5. Scalability

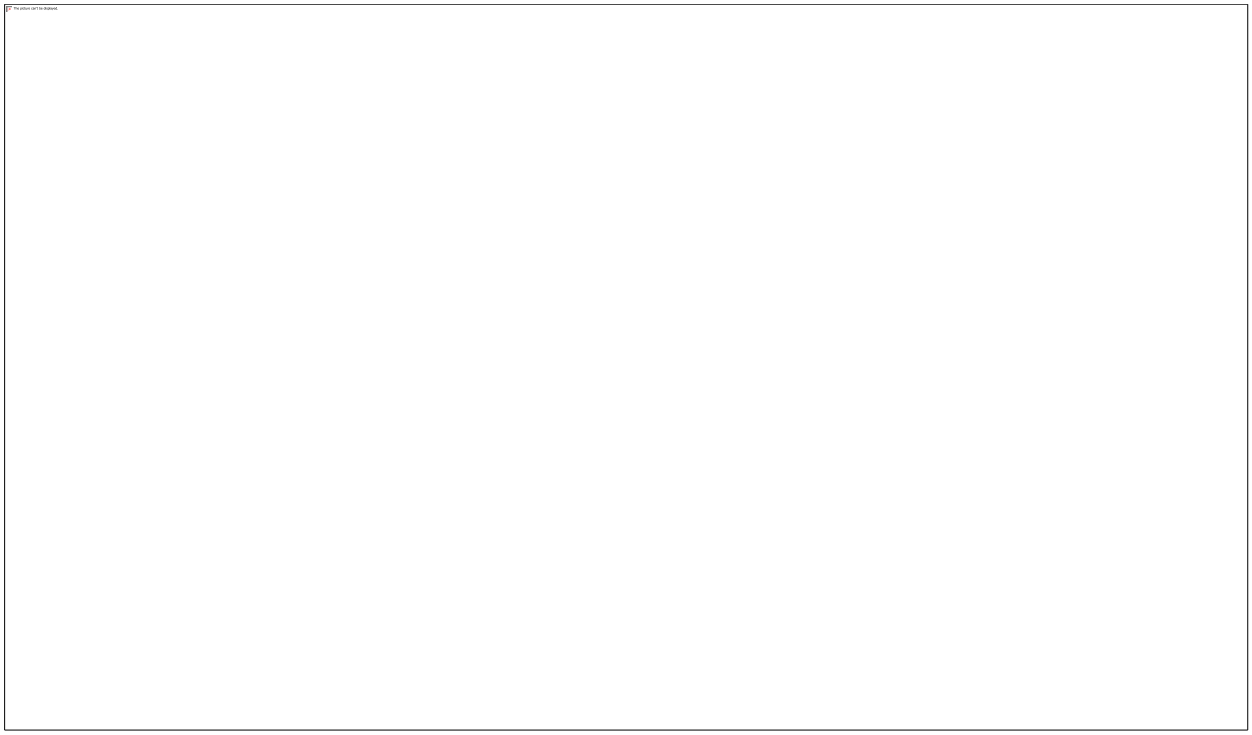

Figure 3.1: Formulating use case diagrams for these 4 functional requirements

Scenario: Given a system that handles sensitive data, find a suitable way to make this system secure to prevent any unauthorized access to information.

Since a component of the application was being dealt with, the main users for the application will not be only the users of the application. It will be the people who are building the application. They are the main users of the project. Apart from that, other users that were identified include the nurses and doctor who access the mobile application. This was obtained because of the requirements drawn to tackle security. As a result, the identified users include:

1. System Developers
2. Individuals in charge of the assigned mobile devices for MHealth

Looking at the Applied project and what is being dealt with, stakeholders of the database include the supervisors of the project. Encryption is the most effective way to achieve data security. To read an encrypted file, you must have access to a secret key or password that

enables you to decrypt it. Unencrypted data is called plain text, and encrypted data is referred to as cipher text.

3.2 Encryption

The two forms of encryption discussed in the related works are end-to-end encryption and secure channel encryption. End-to-end encryption refers to encrypting data before the data is sent or received. If data is encrypted before it enters a channel, the channel does not become a problem or is not even considered, even if the channel is not secured. According to *Whatsapp*, its end-to-end encryption ensures that only the person being communicating with can read what is sent, such that nobody else in between can read it, including WhatsApp itself (Whatsapp.com,2017). Messages are secured with a lock, and only the recipient and sender have a special key needed to unlock and read the message. For added protection, every message sent has a unique lock and key. Moreover, all of this happens automatically; there is no need to turn on settings or set up special secret chats to secure one's messages.

The second approach is secure channel encryption. This involves encrypting the channel through which information is being transferred. Some of these include *https* and SSH(STTP). As the research is goes on, these two forms of encryption will be considered to determine which can be implemented to enhance the security of the database.

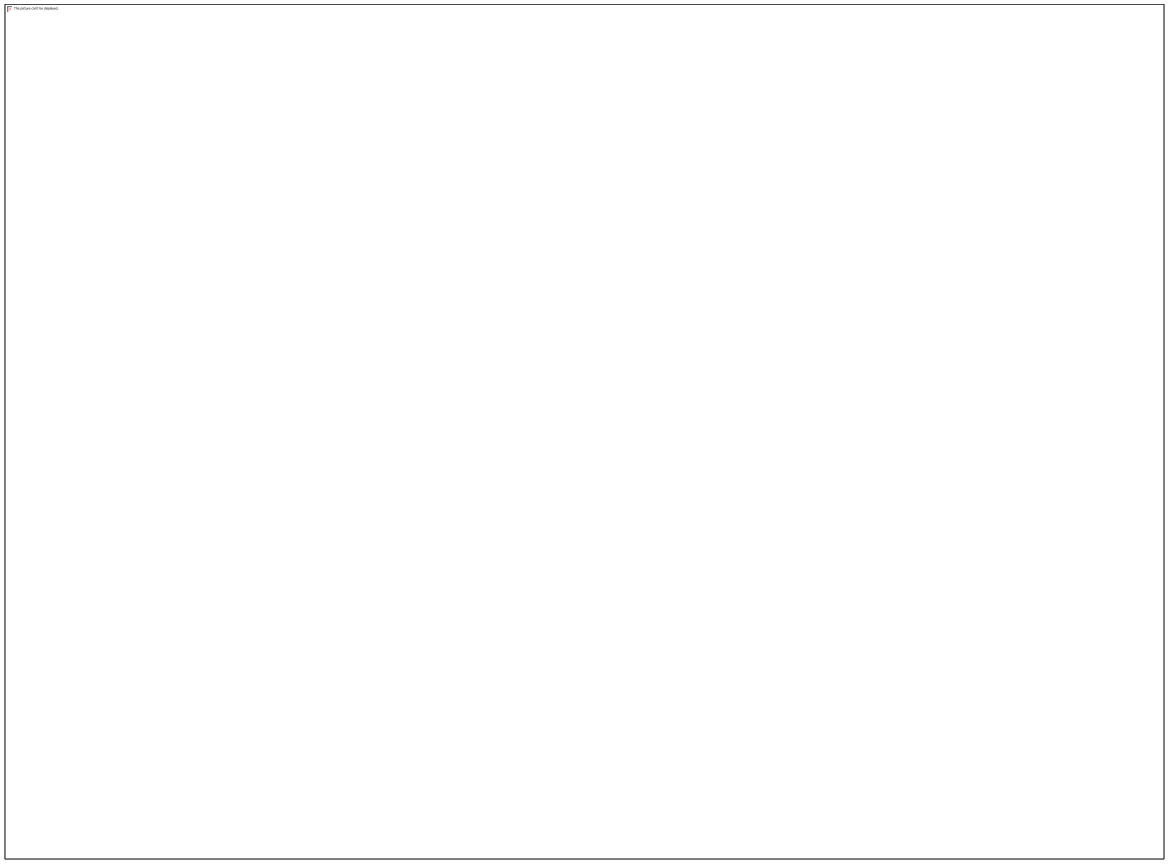

Figure 3.2: Sequence diagram for encryption

Since both the web application and smartphone device will be using end-to-end encryption, this will have to be implement this in both java and php. This project however focuses on the java aspect. This method, end-to-end encryption, means that the data will be encrypted before it is sent over the network for the doctors to receive it. And then, when the doctors open the application, the data will be decrypted once again for use.

Figure3.3: Proposed encryption method

The proposed method of encryption is the AES method. This is due to its symmetrical nature.

3.3 Password Authentication/Access

For password authentication and access, we will be looking at ways of preventing access to the database without the necessary credentials. Some people may mistakenly give out their credential; as such, there must be mechanisms to prevent unauthorized access after has happened.

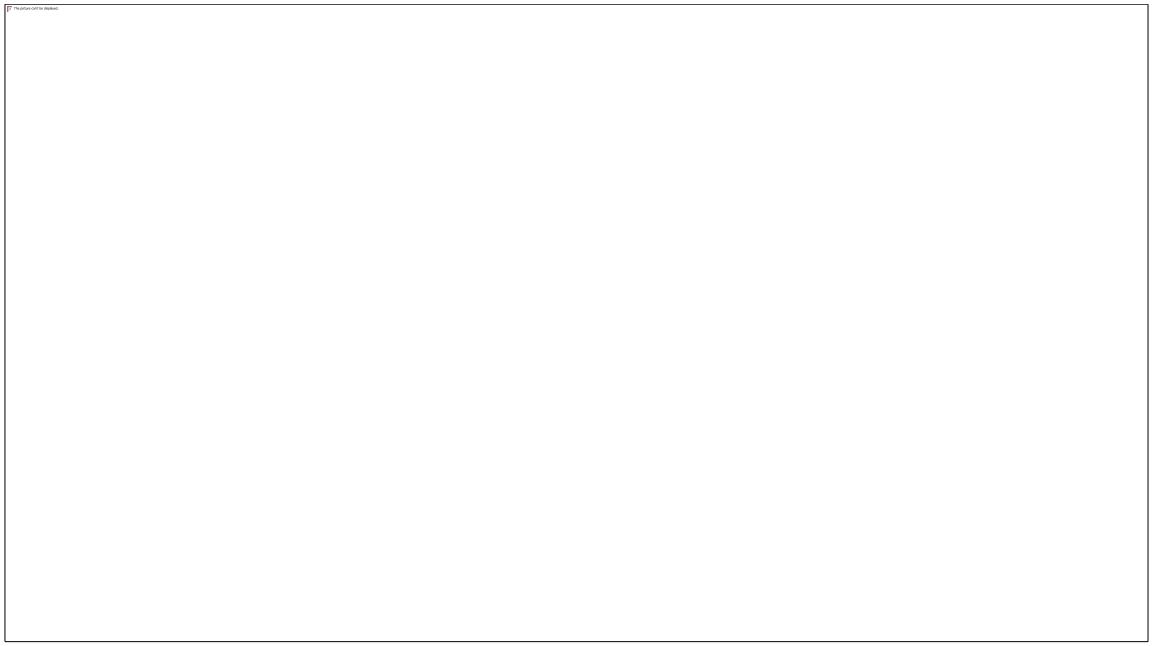

Figure 3.4: Sequence diagram for Password Authentication

3.4 Two-Factor Authentication

This functional requirement is the last security measure required to prevent unauthorized access into the application. After the user has entered their credentials, that is their password and username, there will be a pop-up dialog box that will ask for their sub district id to confirm their identity. This is the sequence the sequence diagram for this functional requirement

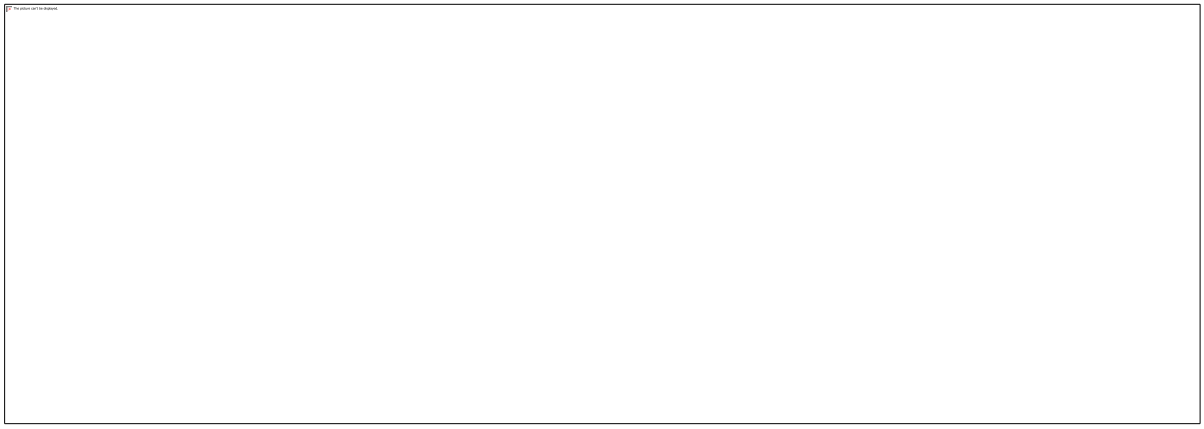

Figure 3.5: Sequence diagram for Two-factor Authentication

3.5 Conclusion on Requirements

The results obtained from the research and interviews served as a yardstick to map out the desired functional requirements. Sequence diagrams aided in mapping out the entire process of interaction between the user and the application. This facilitates a clearer picture of how things with work on a higher level. The functional requirements were molded through a sequence diagram that provided a scenario to map out the security needs of the application. These functional requirements include data encryption, password authentication and two-factor authentication. In the next chapter, the architecture and design of the system to solve the security needs of the mobile application will be addressed.

Chapter 4: Architecture and Design

Evaluating the requirements given the scope of this project will include just three of the stated requirements. These are:

1. Encryption of database
2. Password Authentication
3. Two factor authentication

In this chapter, the various ways by which the stated functional requirements are going to be achieved will be discussed. In view of this, it is only right to state that, the MHealth application, ***"Yaresa"*** has already been developed and is operational already. The purpose of this chapter is to line out the plan for implementing of the functional requirements which aid anyone to understand what is going to achieve the functional requirements. The high-level system is responsible for providing a suitable paradigm through the implementation of various functions and processes to complete the activities of functional requirements. Suitably, each functional requirement will require high-level architecture that will serve as a yardstick to aid the developer in achieving the functional requirements at hand.

4.1 Encryption of Database

When it comes to encryption of any data that must do with the mobile application, the most important and paramount entity of concern is the database. Encryption of the database is very important to maintain the integrity of data for the application and to prevent theft of information. Before any encryption algorithm is implemented, a description must be given of when, where and how the encryption should happen. Knowing where the database is stored is fundamental. Also, the type of database being used must be known. The mobile application uses SQLite database. Before delving into the libraries and encryption methods, let us look at how the encryption of the database will take place. This is the description of the high-level design:

1. When the application starts for the first time, a passphrase is used to encrypt the database.
2. Apart from this, anytime the application is started, the database is decrypted using the passphrase.

3. After this, any transaction can happen, that is, updating the tables, users and any other relevant transactions.
4. When the user closes the application, before shutting down, the application quickly encrypts the database. This will prevent anyone who has access to the database outside the application from using the information if they don't have the passphrase.

4.2 Password Authentication

Password authentication is important because the mobile application has just one slot for user access, and this is not secure. As a result, people can access the data through the application. To make the application more secure, there will be a password box in addition to the username, to ensure users have their usernames and passwords requested before accessing data that is involved with the mobile application.

The high-level architecture design for this functional requirement is straight forward since it will be the first thing the user interacts with when they open the mobile application. the following points are the steps for the interaction when it comes to password authentication.

1. User enters username and password
2. Credentials are vetted to see if they are present in the database table that keep all usernames and passwords
3. If the credentials check out, the user is granted access to the mobile application

4.3 Two-factor Authentication

This provides an extra layer of security, which tends to increase the security of the mobile application. Two factor authentication is a system whereby apart from using your traditional credential, that is, username and password to login, another piece of information that only the original user knows is required. This makes is more difficult for intruders to steal personal data or identity. This reduces the number of identity theft cases since now intruders will require more than the user's username and password (SecurEnvoy, 2017). From 2011, Google started using

two-factor authentications for their online users, and this move was replicated by MSN and Yahoo (Arthur, 2011).

The high-level architecture design for this implementation will be as follows:

1. The user enters their username and password
2. The user is asked for their second token which will be unique for every user
3. The user is granted access to the application

Chapter 5: Implementation

5.1 Background of Implementation

In this chapter, in-depth look will be taken on how the various function requirements were tackled. This chapter will provide a high-level description of the implementation including tools, libraries, frameworks, APIs and components.

5.2 Encryption of Database

The implementation of this functional requirement involved the addition of library, overwriting of some methods and the deletion of some imports.

5.2.1 Addition of Library

The library necessary for the encryption of the database is an open source library called ***"SQLCipher"***. This is an upgrade of the already-existing Android library for the SQLiteDatabase, but with the encryption phase added. SQLCipher is a unique build of the original SQLite database that performs encryption while the mobile application has been started. SQLCipher still implements the stock SQLite API the manage tables in of the database with the use of Structured Query Language(SQL). In the background of operations, SQLCipher manages the security of the

database, ensuring that data is encrypted and decrypted as it is written and read from memory (Zetetic, 2017). It was developed by Zetetic LLC.

On implementation, there were 3 main processes that needed to be done to ensure the library was correctly implemented on the ***"Yaresa"*** mobile application. The following three subsections will discuss these processes.

5.2.2 Updating of Dependencies

In the Gradle, integration can be done by adding the second line in the diagram to the dependencies section of the *app/gradle* file:

```
dependencies
{
      compile fileTree(dir:'libs',include: '.jar')
      compile 'net.zetetic:android-database-sqlcipher:3.5.6@aar'
}
```

5.2.3 Importing the Library

Now we will need to import library into the various classes that are going to implement the library. This is paramount to initializing native libraries for the SQLCipher. This is *'android.database.sqlite.SQLiteDatabase'* to *'net.sqlcipher.database.SQLiteDatabase.'*. Also, the call to *SQLiteDatabase.loadLibs(this)* must occur before any database process.

5.2.4 Overwriting of Methods

For the overwriting of methods, there were two methods that needed to be overwritten, and these are the getReadableDatabase() and getWritableDatabase(). These methods are responsible for writing and reading from the encrypted database. They must be

overwritten since they need a passphrase to perform reading or writing into a database. This passphrase is declared in the beginning of the creation of the constants.

```
private static final String PASSPHRASE="mhealthprotection";


//Declaration of passphrase


public SQLiteDatabase getReadableDatabase() {

   return(super.getReadableDatabase(PASSPHRASE));
}


public SQLiteDatabase getWritableDatabase() {
   return(super.getWritableDatabase(PASSPHRASE));
}
```

5.2.5 End of Encryption

After these steps have been completed, the application can be compiled and then installed on to a mobile device for testing. It must be noted that if the application has already been installed, it should be reinstalled, by uninstalling then installing again. This will aid in creating the database properly and preventing the mobile application from crashing.

5.3 Password Authentication Implementation

In this section, implementation must do with adding to things to the application. A column was added to the CHO table, which consisted of the username of the users. This new local variable was cho_pass which was of String nature to store the password of users. Also, a new textbox was added to allow the entry of passwords by the user.

Figure 5.1: Table indicating new field cho_pass

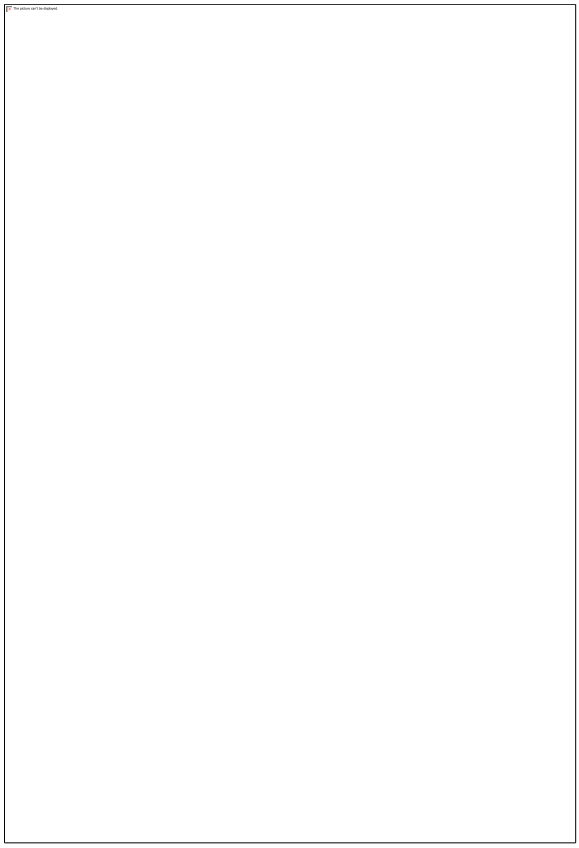

Figure5.2: Virtual device showing the new field for the password

After these two steps were completed, the database was populated to enable the application access the resources of the application. As a result, one user was added to test the application.

For the code, a new dialog box was inserted for the password. Also, a new local variable was added to the cho class to enable all chos to have a password section. Code was also added to

the loginAndStart() function to facilitate the authentication through the username and password section.

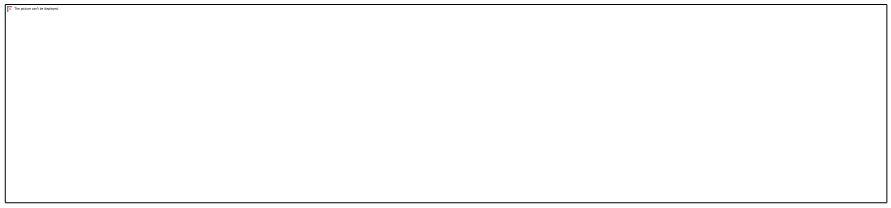

Figure 5.3: Populating table with the new user with a password

Another thing that was added was a forgotten password section to aid users who have forgotten their password obtain a new password by filling in their e-mail credentials. A new class and table was added with just the user's id, password and email to ensure separate information and prevent a probable sql injection attack. As a result, an email was used to verify the user and reset a password to enable the user login.

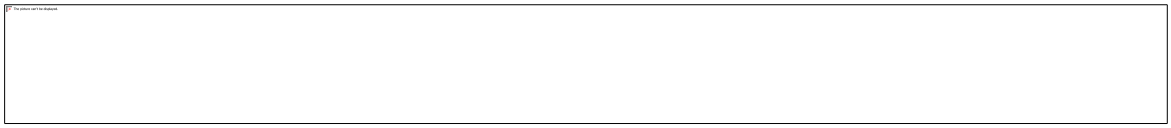

Figure 5.3: Populating table cho_fPassword with one user and their password.

Figure 5.4: Dialog bog requesting user email to reset password.

5.4 Two-factor Authentication

The two-factor authentication was implemented by adding a pop dialog box that shows up after the user has entered their username and password. The last token that will be asked for is their unique token which will be the column subdistrict_id. With this, the user will be given access to the mobile application.

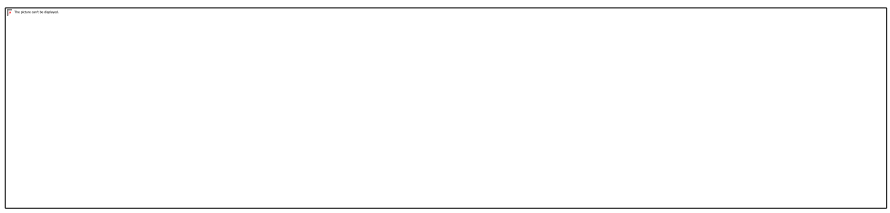

Figure 5.5: Table showing the second security token sundistrict_id

For the two factor authentication, code was added in to LoginAndStart() method in the main activity of the application. This will authenticate the user by bring up a dialog box which will ask for the user's security token. After this, has been checked, the system will verify with the data base and either proceed to let the user in to the system or prevent the user from accessing the database.

Also, no new table was added to the database to verify the user. The subdistrict_id was used to verify the user.

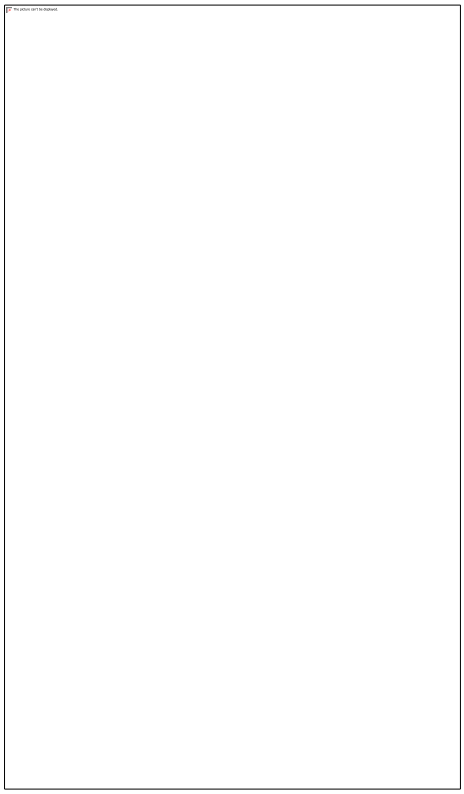

Figure 5.6: Yaresa application showing dialog box requesting to subdistrict_id

5.5 Specific Algorithms and Approaches for MHealth

Various methods had to be implemented to achieve the encryption of the databases. The first thing that was done was the addition of tables that were not created when the new encrypted database was created. As a result, the system tries to create the database, if it doesn't exist it creates it. All the table names and queries were stored in a hashmap. As a result, the tables and insertions that were missing were all created.

For the password authentication and two factor authentications, no new method was created. The data had to be fetched from the database and be compared to the data entered by the user.

Another method that was explored for the database was migrating the unencrypted database's contents to a new database that was encrypted but this came with problems and a result was not implemented.

For reading and writing into the encrypted database, the method getReadable and getWritable we tweaked to add the password of the database to it.

5.6 Conclusion of Implementation

In conclusion, the password authentication, two factor authentication and database encryption were successfully implemented as planned. In the next chapter, we will be evaluating test results given various scenarios. This will help guide us to see if these functional requirements have been achieved. They will be handled in various subsections of the next chapter according to each functional requirement. Each functional requirement will have its own test and result analysis.

Chapter 6: Testing, Experimentation and Result Analysis

6.1 Background of Testing, Experimentation and Result Analysis

Evaluating the functional requirements that have been addressed and solved, this chapter is mainly directed toward the testing of these various functional requirements to see if they have been fully achieved, and what suggested methods will be implemented to solve these functional requirements. Also, there will be evaluation of non-functional requirements. Since there are various functional requirements, the non-functional requirements will be evaluated every time a functional requirement is added. The functional requirements that will be evaluated include the following:

1. Encryption of database
2. Password authentication

3. Two factor authentication

The following sections will address the testing, experimentation and results analysis of the functional requirements listed above. A few non-functional requirements will also be evaluated, to aid in determining if the additions to the system make the system more efficient. These non-functional requirements include:

1. Data Integrity
2. Security
3. Performance
4. Stability

6.2 Testing, Experimentation and Result Analysis of Encryption of Database

For the encryption of the database, there were two instances for which the functional requirement was tested for. The nature of the database and its contents were evaluated under these instances:

1. Pre-Encryption
2. Post-Encryption

Resources that were used to aid the testing of this feature include SQLitestudio, Android Studio, atom IDE and the adb resource on the terminal.

6.2.1 Steps for the testing

For the test of the functional requirement under both instances, the following generic steps were involved in evaluating the resources that were needed for the result analysis.

1. Run the ***"Yaresa"*** application, with a virtual device or a rooted android device, then retrieve unencrypted database using the adb shell in terminal. This is done outside the adb shell.

```
Nathans-MacBook-Pro:~ nathandonkor$ adb pull /data/data/com.ashesi.cs.mhealth.da
ta/databases/mhealth "/Users/nathandonkor/Desktop"
[100%] /data/data/com.ashesi.cs.mhealth.data/databases/mhealth
Nathans-MacBook-Pro:~ nathandonkor$
```

Using the ***adb pull*** function to retrieve the database, like from the screenshot above, if you are successful you should see something similar. The function is,

```
adb pull
/data/data/com.ashesi.cs.mhealth.data/databases/mhealth
"/Users/nathandonkor/Desktop"
```

2. After this, the data in the database is unencrypted since it can be viewed as readable human text using the atom IDE or SQLitestudio.

Figure 6.1: Readable database in atom IDE.

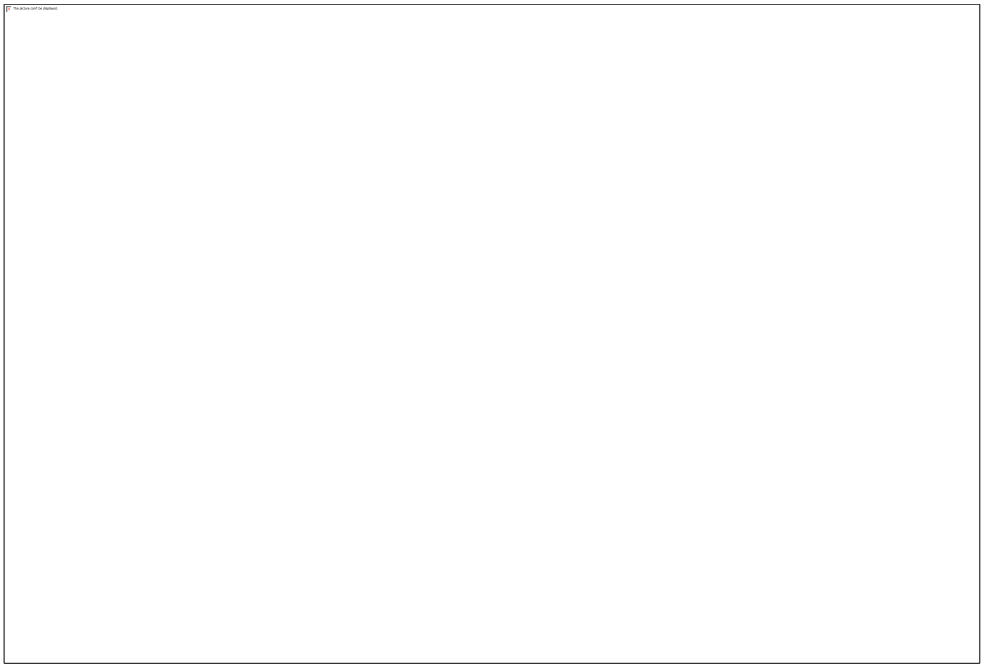

Figure 6.2: Unencrypted database view

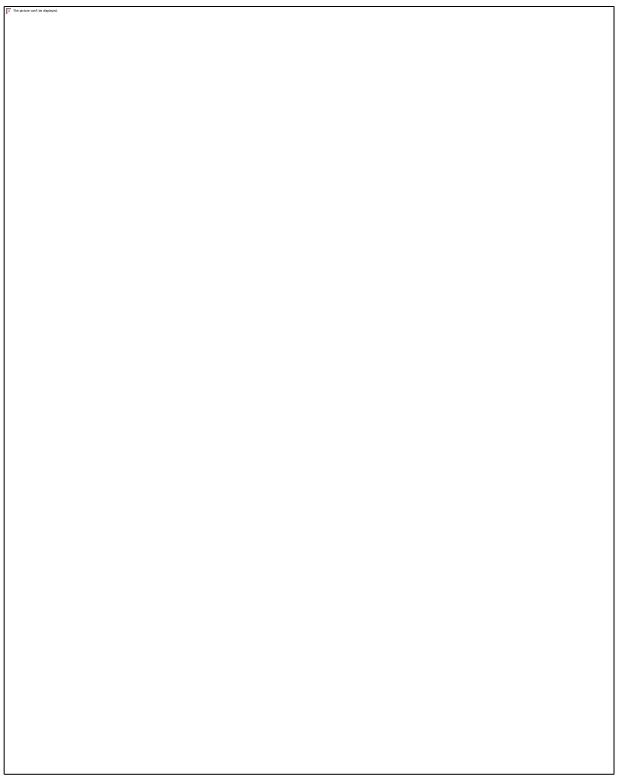

Figure 6.3: Dialog box in SQLiteStudio for unencrypted database with no password.

3. After the library has been added to the program and all necessary methods and imports have been done, the application will be restarted to obtain the database again
4. After the database has been retrieved, we will now realize that, when trying to view the database, it no longer shows as readable data, unless we use SQLitestudio and enter the password for the database to decrypt the data in the application.

```
Nathans-MacBook-Pro:~ nathandonkor$ adb pull /data/data/com.ashesi.cs.mhealth.da
ta/databases/mhealth "/Users/nathandonkor/Desktop"
[100%] /data/data/com.ashesi.cs.mhealth.data/databases/mhealth
Nathans-MacBook-Pro:~ nathandonkor$
```

Using the ***adb pull*** function to retrieve the database, like from the screenshot above, if you are successful you should see something similar. The function is,

```
adb pull /data/data/com.ashesi.cs.mhealth.data/databases/mhealth
"/Users/nathandonkor/Desktop"
```

Figure 6.4: Encrypted database viewed using Atom IDE

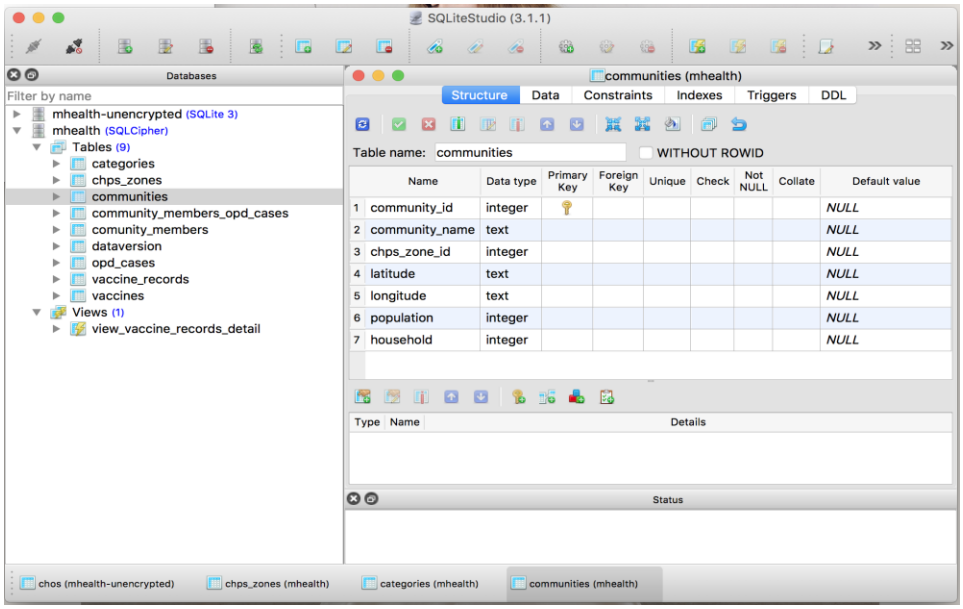

Figure 6.5: Encrypted database view after entering password

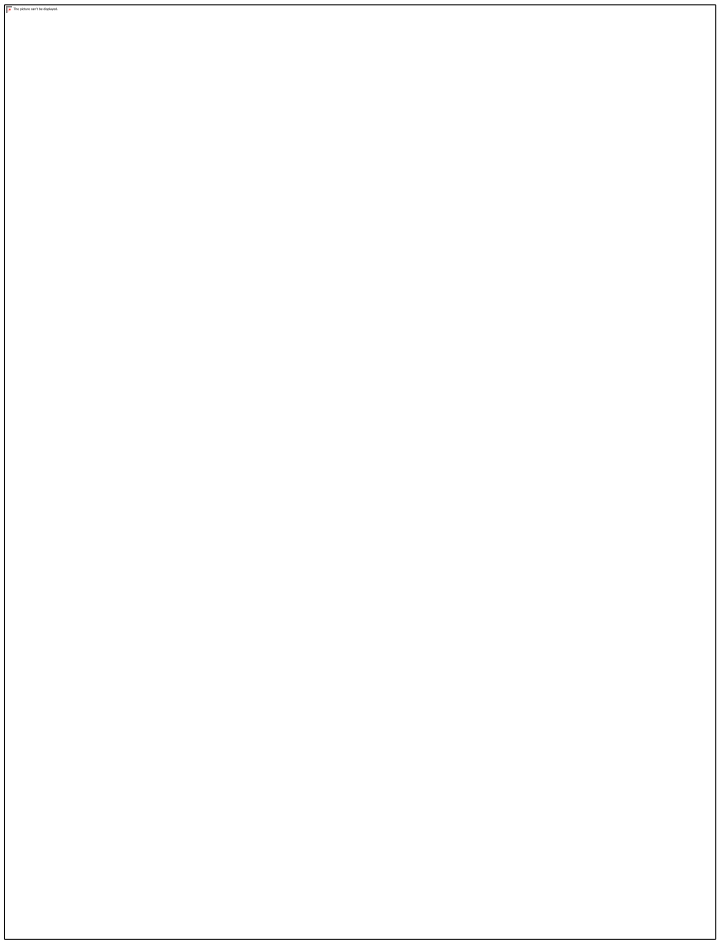

Figure 6.6: Dialog box in SQLiteStudio for encrypted database when entering the passphrase.

6.2.2 Evaluating Non-Functional Requirements

5. Data Integrity – Evaluating the integrity of the data, the data in the database remained intact. Tables and their contents remained intact. The size of the database is relatively the same after the encryption.
6. Security – The data is now 100% secure. There is however one caveat; if anyone has access to the passphrase, they can access the database.
7. Performance (Response time) – The application could still perform tasks in the same time. There was no hesitation or freezing when booting the application.
8. Stability – The application remains stable after adding the encryption phase.

6.3 Testing, Experimentation and Result Analysis of Password Authentication

Testing for the password authentication involved interacting with the user interface; that is, entering various combinations of usernames and passwords. Some of these combinations include:

1. Correct username, no password
2. Correct username, wrong password
3. Correct username, correct password
4. No username, correct password
5. Wrong username, correct password
6. No username, no password

Looking at these scenarios, it only implied the username and password were being looked up from a database which had user information, that is both password and username, amongst other details necessary for logging in. Thus, with these scenarios, if the username and password are correct, the user will have access to the resources of the ***"Yaresa"*** mobile application. The other scenarios provided messages indicating either the password or username were incorrect.

6.3.1 Evaluating Non-Functional Requirements

7. Security – User access is now limited only to users who have their names enrolled in the database. As a result, the mobile application is more secure since there is not just one user now (“Admin”). It has multiple users who also have corresponding passwords to grant them access to the resources of the mobile application.
8. Performance (Response time) – The application could still perform tasks within the same time; there was no hesitation or freezing when booting the application. The application moves to the main screen as soon as the user submits his username and password.
9. Stability – The application remains stable after adding the password field and adding the new users.

6.4 Testing, Experimentation and Result Analysis of Two-factor Authentication

For the testing and experimentation of the two-factor authentication, since only one field was chosen to be worked on as the security token, testing this category was straightforward. We just had to enter strings of the field required that were in the database and then strings that were not in the database.

6.4.1 Evaluating Non-Functional Requirements

10. Security – Adding another field for authentication makes it easier for social engineers to try and get access to files or the application resources.
11. Performance (Response time) –The application could still perform task in the same time; there was no hesitation or freezing when booting the application. The application moves to the main screen as soon as the user submits the last security token.
12. Stability – The application remains stable after adding the two-factor phase.

Evaluating the implemented functional requirements and looking at the Ministry of Health

document, some of the expectations of the documents have been accomplished. Data Storage, internal use of data, data access by subjects and disclosure to third party have all been achieved. For data storage, we see that the data is being stored in the application and at the same time is secure. Internal use of data and data access by subjects are achieved because of using authentication methods.

Chapter 7: Conclusion

7.1 Challenges

Upon completing this project, despite the success, it also right to acknowledge the occurrence of challenges in this project. These include:

1. Learning Android: Since it was an individual project and Android development is not part of my skill set, it took some time to get used to the Android environment. Also, since there was a time constraint on the project, this put some significant amount of pressure to deliver on the project. It was challenging, but in the end, I could complete the tasks at hand.

1. Integration of SQLCipher into the mobile application: This was a challenge because of the already existing database. There was a need to recreate the tables and populate them using the SQLCipher. Although this seemed like a simple task, a lot of errors came up but with the help of my supervisor and online materials, it was solved.

2. User authentication: This was also a challenge since the interaction with the database had changed due to the new library introduced. Authentication was a problem because sometimes it was difficult to obtain the data from the database.

3. Versioning Problem: since the databases were set up in a peculiar manner, this made it difficult for me to create some of the tables. As a result, I had to form methods to redo creation of some tables and also some insertions to make the database fully useful again. Another solution that was a explored was converting the already existing database to the encrypted one, at this stage to we faced file directory problems.

7.2 Further Work

At the end of this project, given the time constraint, not all functional requirements were achieved. Considering this, if there is going to be any work concerning security on the mobile application, there are a few suggestions that can be implemented to ensure the security of the mobile

application is top-notch. These suggestions are also functional requirements that were derived from the research conducted on this project. They include:

4. Encryption of backup
5. Keeping patches current
6. Avoiding the use of third party applications
7. Avoiding the use of a shared server
8. Enabling security controls
9. Encryption of transmission routes

References

Arthur, C. (2011). *How Google, Facebook and Hotmail aim to stop holiday hacking*. *the Guardian*. Retrieved 17 April 2017, from https://www.theguardian.com/technology/2011/aug/05/google-facebook-hotmail-stop-hacking

Best Practices for Database Security. *(2016).* Applicure.com. *Retrieved 17 October 2016, from http://www.applicure.com/blog/database-security-best-practice*

Health Matrix Network. (2008). *Legal and Policy Framework for Health Information and Health Data Reporting.* Ghana Ministry of Health.

Mueller, R., Schrittwieser, S., Fruehwirt, P., Kieseberg, P., & Weippl, E. (2014, December). What's new with whatsapp & co.? Revisiting the security of smartphone messaging applications. In *Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services* (pp. 142-151). ACM.

Rottermanner, C., Kieseberg, P., Huber, M., Schmiedecker, M., & Schrittwieser, S. (2015, December). Privacy and data protection in smartphone messengers. In *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services* (p. 83). ACM.

Rumsby, B. (2016). *US superstars Serena and Venus Williams and Simone Biles given drugs exemption, Russian hackers reveal*. *The Telegraph*. Retrieved 17 April 2017, from http://www.telegraph.co.uk/sport/2016/09/13/us-superstars-serena-and-venus-williams-and-simone-biles-given-d/

Sadeghi, Ahmad-Reza. "Mobile security and privacy: The quest for the mighty access control." *Proceedings of the 18th ACM symposium on Access control models and technologies*. ACM, 2013.

What Is Mobile Security? Webopedia Definition. (n.d.). Retrieved from http://www.webopedia.com/TERM/M/mobile_security.html

What is Two Factor Authentication? | SecurEnvoy. (2017). *Securenvoy.com*. Retrieved 24 March 2017, from https://www.securenvoy.com/two-factor-authentication/what-is-2fa.shtm

WhatsApp FAQ - End-to-End Encryption. (2017). *WhatsApp.com*. Retrieved 24 March 2017, from https://www.whatsapp.com/faq/en/general/28030015(2017). *Moh.gov.gh*. Retrieved 24 March 2017, from http://www.moh.gov.gh/wp-content/uploads/2016/02/Policy-and-Legal-Framework-for-HMIS.pdf

Zetetic (2017). *Zetetic.net*. Retrieved 24 March 2017, from https://www.zetetic.net/sqlcipher/sqlcipher-for-android/

Lecture 20: Mobile Phone Security | Lecture Videos | Computer Systems Security | Electrical Engineering and Computer Science | MIT OpenCourseWare. (2017). *Ocw.mit.edu*. Retrieved 1 May 2017, from https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-858-computer-systems-security-fall-2014/video-lectures/lecture-20-mobile-phone-security/

Appendix

Appendix 1 – Ministry of Health's policies on exchange on health information

The Ministry of Health shall determine the best way to allow individuals to participate in and consent to electronic health information exchange.

1. The Ministry of Health shall implement adequate security measures for protecting health information. This shall include techniques for authenticating requesters of health information, implementing proper access controls and maintaining adequate audit trails for monitoring access to health data Other legal provisions to be considered
2. Data Collection: In line with the principles of information privacy, data collected by the health sector shall be non-patient identifiable. This shall be different from the policies on medical records management.
3. Data Storage: Appropriate standards are needed in relation to the condition in which the data is maintained. This includes precautions against fire and other accidents and criminal acts. In the case of computer-based records, the additional question arises as to how the records can be accessed. Because of data sensitivity, appropriate security against unauthorized access and modification is essential.
4. Internal Use of Data: Medical data should only be used for the purposes for which it was collected, and for additional purposes authorized by law, or consented to by the data subject. The purposes for which health data is collected needs to be clear.
5. Disclosure to Third Parties: Since medical data is sensitive, and since a duty of confidence generally applies to data which a health care professional gathers during his relationship with a patient, it is necessary to regard healthcare data as being unavailable to third parties in the absence of a clear and authoritative reason. In the case of a referral, care is needed to ensure that only relevant parts of the patient's history are communicated.
6. Data Access by Subjects: The principle of data ownership is to appreciate that, while the records (the documents or disks) are unequivocally the property of the practitioner or

institution, the data is not. Data is not capable of being owned, and many different people have an interest in it, including and especially the person to whom it relates.

7. Record Transfer: Although records are owned by their originator, a patient has a very real interest in having them, or at least an accurate representation of their contents, transferred to his new health care professional. The practice of transferring records when an appropriately documented request is made is therefore highly desirable from a treatment viewpoint.
8. Record Destruction: Patient history is one of the relatively few classes of record for which some genuine justification exists for long-term retention. However, the volume of information which is generated becomes very large, and much of it becomes irrelevant over time, and hence periodic summarization and destruction of old material should be aimed at.

Source: (Health Matrix Network, 2008)

Appendix 2 – Questions Asked During Interviews

1. What comes to mind when we refer to mobile security?
2. Is it important to have security for mobile applications?
3. What must be protected at all costs?
4. What are the various ways by which a mobile application can be protected?